(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 841 624 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
13.05.1998 Bulletin 1998/20

(51) Int. Cl.[6]: **G06F 17/28**

(21) Application number: **96650050.6**

(22) Date of filing: **08.11.1996**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV RO SI

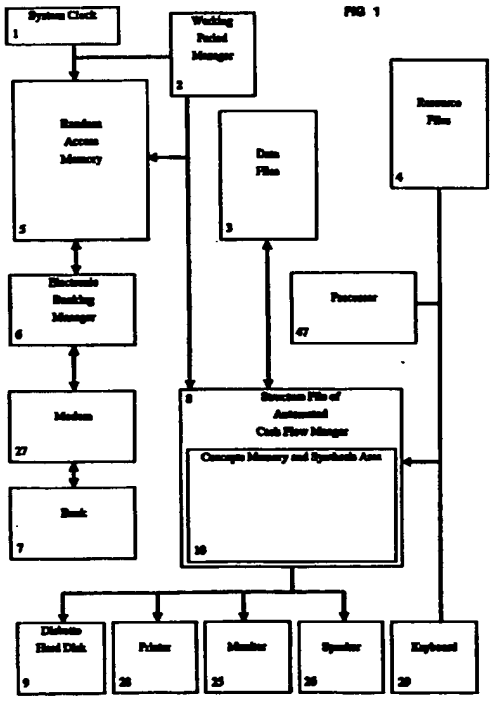
(71) Applicant: **Softmark Limited**
County Cork (IE)

(72) Inventor: **Nordstrom, Rabbe, Dr.**
County Cork (IE)

(74) Representative:
Weldon, Michael James et al
c/o John O'Brien & Associates,
Third Floor,
Duncairn House,
14 Carysfort Avenue
Blackrock, Co. Dublin (IE)

(54) Input and output communication in a data processing system

(57) A processor (47) uses a concepts memory area (10) to interpret received inputs such as invoices and to generate coherent output communication in text or fluent speech. The processor (47) uses concepts, each being a part of program code and a linked basic communication element. A parallel recursive function matrix manager (10(a)) is used in processing of the concepts and reference is made to a conception, learning, associative utilisation manager (50) to assist in understanding inputs and developing outputs. The basic communication elements may be triggered for associated speech resources (24), retrieved and processed downstream.



EP 0 841 624 A1

Description

The invention relates to communication in a data processing system of the "intelligent" type such as an expert system. Such a system may, for example, be based on the neural network principle and may be exclusively rule-based or may include a case based reasoning (CBR) system.

The purpose of intelligent systems is to learn or adjust to a multitude of inputs or stimuli and allow a multitude of possible results, i.e. there may be a number of "right" results. These results cause events to occur such as the automatic transmission of a signal to a financial institution to cause lodgement of money to an account. Development of such systems has progressed to the stage where some can operate in a "hands off" mode for considerable periods of time.

However, such features are of little benefit if there is inadequate input and output communication. For output communication, in some cases system events can be communicated by display of numbers or numerical functions in a clear manner. However, it is increasingly the case that for the important events to be understood, much more comprehensive communication is required. For example, it is desirable that the system be capable of outputting coherent speech to indicate the events which have occurred, and indeed events which may occur in the future, depending on user inputs. It is also desirable that the system be capable of understanding inputs with much less human intervention, and preferably little or no human intervention.

Heretofore, the approach to generating output communication has been to select or build pre-determined messages. For example, in DE4323021 a system accesses, interrelates, and combines prototype patterns to generate a composite image output. In JP07319538 a building monitoring system is described in which audio data is selected and outputted according to abnormal state information. In EP0052757B (IBM) a method is described in which various tables are used by a message build program to display one of a set of pre-determined messages.

In all of these systems the basic principle is operation of a communication means to select portions of messages and to combine them together. This approach is apparently satisfactory for many situations, such as for generating alerts to indicate specific and predetermined events, such as a fault in a diskette door. However, where comprehensive system event information is to be communicated for complex data processing systems it would be necessary to store a very large number of message parts and it would be unworkable to retrieve and process these messages in a meaningful manner. Further, it would be extremely difficult to foresee all possible communication which may arise in the future. The prior systems do not generate coherent communication describing any and every possible system event irrespective of the probability of them arising.

Further, while the prior art includes many systems for reading inputs such as OCR systems, little has been done to integrate this function with the data processing.

The invention is directed towards providing a data processing system and method to overcome these problems.

According to the invention, there is provided a communication method for interpreting received communication and building output coherent communication indicating system events, the method being carried out by a system comprising a processor, a memory, an input device, and an output device, characterised in that :

the processor automatically interprets inputs or generates coherent communication in-line as data is being processed using a communication means linked with data processing program code.

This allows very flexible interpretation of received inputs and generation of effectively unlimited output coherent communication in real time.

Preferably, the processor processes concepts to interpret inputs or generate coherent communication, each concept comprising a communication element linked to a part of program code. Use of such concepts is a very effective way of performing the communication operations - each concept is a unit which may be effectively processed in a simple manner. Thus, while the full output coherent communication may be very comprehensive and complex, the "building blocks" are handled in a simple manner.

In one embodiment, the concepts are processed in one memory area, either actual or virtual, and either time-dependent or time-independent. This is an efficient way of processing the concepts.

Preferably, the processor processes concepts having communication elements linked with program code representing processing results, and concepts having communication elements linked with program code used for interim processing before a result is determined. This means that the output communication may be extremely comprehensive. Heretofore, output communication has taken the form of messages indicating results or final status of events.

In another embodiment, the processor interactively utilises a communication matrix manager providing a parallel recursive function for data comparison to build a communication knowledge base. This manager is a very effective tool which assists in a non-intrusive manner in performing the communication operations. It is particularly useful for communication building as it can constantly scroll through the elements, and at a larger granularity, the concepts.

Preferably, the processor uses a conception, learning, associative and utilisation manager to perform communica-

tion operations, said manager including:-

- a definition of terms manager which performs a full text analysis of inputs to define as many terms as possible, and

- a validation manager which performs validation operations on the defined terms.

This also assists in the communication operations and allows it to be achieved in real time.

In one embodiment, the communication means comprises communication elements which trigger resources including text or speech strings. This allows the communication operations to be performed with little effect on the data processing - the basic communication being interpreted or generated by processing of the concepts and being improved upon by processing of the resources.

Preferably, the system comprises speech resources, a speech resources assembly manager, and a speech resources reference buffer, the assembly manager and the reference buffer performing the steps of:-

- loading blocks of resources to a memory;

- monitoring speech output and determining when resources of a block are nearly utilised;

- loading a fresh block; and

- clearing the previous block when a fresh block is activated.

This is a very simple way of ensuring fluency of output speech and that memory overflow does not occur.

In one embodiment, the resources are used to build coherent output communication which is further automatically edited before output.

According to another aspect, the invention provides a data processing system comprising a processor, a memory, an input device, and an output device, characterised in that the processor comprises means for automatically interpreting inputs or generating coherent communication as data is being processed using a communication means linked with data processing system code.

The invention will be more clearly understood from the following description of some embodiments thereof given by way of example only with reference to the accompanying drawings in which:-

Fig. 1 is an overview schematic representation of an expert system of the invention;

Fig. 2 is a block diagram showing the system in more detail; and

Figs. 3 & 4 are still more detailed views showing the system at a lower level.

Referring to the drawings and initially to Fig. 1 there is shown an expert system of the invention. As illustrated, the system comprises a system clock 1, a working period manager 2 connected to the clock 1, data files 3, a resource file 4, a random access memory (RAM) 5 and an electronic banking manager 6. A bank system 7 communicates with the electronic banking manager 6 via a modem 27. The bank system 7 is not part of the expert system 1. The system also includes one or more processors 47 which may be of any suitable type such as that of a stand-alone microcomputer, or of a client server. The system comprises a structure file 8 of an automated cash flow manager which uses a memory area 10, referred to as a concepts memory and synthesis area. Although shown separately, the area 10 is part of the RAM 5. The system also comprises disk drives 9, a monitor 25, a speaker 26, a printer 28 and a keyboard 29.

At the purely hardware level, the system is conventional and indeed the method of the invention may be carried out by a wide range of hardware systems when suitably programmed.

The structure file 8 is shown in detail in Fig. 2 in combination with the memory which it utilises. For processing of signals or stimuli which arise within the system there is a pattern assembly manager 11 and a pattern conclusion manager I 20. For processing of stimuli which are received from an external source there is a pattern recognition manager 31, a pattern simulation manager 48, and a pattern conclusion manager II 49. The series of managers 11 and 20 and the series of managers 31, 48 and 49 are connected to a conception, learning, and associative utilisation (CLAU) manager 50. The manager 50 comprises the following modules :

51 - Definition of terms manager. This performs a full text analysis of variable electronic input devices in order to define as many terms as possible, either by comparing those terms with formerly memorized keywords or by selecting new terms within a predefined range of probability.

52 - Validation manager. This takes the defined terms of the manager 51 and preforms a validation control.

53 - Analogy building manager. This determines if an identified term has some meaning for the system's control process. For example, if the system is able to identify some terms like value added tax (VAT), "increase" and some numeric specification like "18%". As a consequence, the analogy building manager first controls the current settings for value added tax, will recognize that there is some change (in the sense of an increase) and will state, that there could be some change for other VAT specifications. This will be presented to the user as a question - if the increase is real and if this has any consequences for the system settings.

54 - Review specification manager. This determines if an identified term has been used in a different or similar context during the past. If so, the system analyses under what circumstances this term has been used and what decisions have been made that time. According to this, the system can decide the same way once more or not or just refer to the past experience.

55 - Synonym and acronym manager. This searches within a specific file for synonymous terms, in order to find out if the selected term has a corresponding item which can be used within the communication building process. This is of importance if the system uses preconfigured speech resources, which are addressed by pointers. If there is no predefined speech resource for the original item for which the system can identify a synonymous term, the pointer is set on the corresponding speech resource specification.

56 - Confusion of ideas manager . The confusion of ideas manager has to evaluate if the selected terms have any correlation to each other or if the plausibility is too low, in order to build the communication process out of the terms found.

57 - Associative manager. The associative manager compares different terms with each other, regarding each term's surrounding context. This means, that a term which the system has found to be strictly used in the business vocabulary, usually will not be mentioned in the communication building process for e.g. medical discussion.

58 - Simulation manager. The simulation manager is needed for only exogenous incitements, for example by the automatic bank report analysis. If a corresponding record, for example in the invoices file, is found, the system doesn't stop at this point. It continues to search for other possibilities, not taking the first choice as the obviously best one.

59 - Incoherence manager. The incoherence manager has to check if the words and sentences which are queued up by the communication building process have sufficient correlation to each other, so that the result makes sense to a human being.

60 - Plausibility manager. The plausibility manager is active in the case of endogenous as well as in the case of exogenous incitements, in order to distinguish between two or more possible answers and, if possible, to decide for the right choice. On the other side, the plausibility manager also takes control for example in the debit entry analyser 33, where the total sum is compared with the amounts registered in the past. If there is some suspicious deviation, the system generates a corresponding comment.

All of the above modules operate in the concepts memory area in which data processing takes place to determine events.

The output from the concepts namely area 10 is fed to downstream modules to complete building of coherent output communication indicating the system events. These include a speech resources assembly manager 21(c) which utilises speech resources 24 and feeds into a speech resources reference buffer 23(b). The buffer 23(b), a prosodic control manager 23(e) and a grammar rules manager 23(d) all feed into a speech editing supervisor 23(c), which in turn controls the speaker 26.

For output of text communication, the output of the concepts memory area 10 is fed into a text communication manager 21(b), which in turn feeds into a document text storage buffer 23(a), in turn driving the monitor 25. In addition, there is a management information system manager 61 which receives data from the concepts memory area 10.

Referring now to Fig. 3, those parts of the structure file for operation in response to internally-generated stimuli or endogenous incitements are illustrated. There is an obligation status manager 13 connected to the pattern assembly manager 11. The pattern assembly manager 11 comprises:

14 - Recursion manager. The recursion manager calculates the optimized result, regarding a variable number of combinable variables. With regard to cash management this means that the manager tries to implement a number of obligations within a number of accounts by using the method of recursive processing. The system selects a certain obligation and one of the defined accounts and controls, if this combination fits. Then it starts incrementing the obligations amount and tries to combine the different obligations with regard to the given account status. When the manager reaches an optimum result, the specified variables are memorized and the search is done for the next account. At the end, the system decides for the highest degree of optimization and selects the most opportune solution. The recursion manager also takes into consideration the available funds and defers the remaining items to the subsequent term of payment.

15 - Due date manager. The due date manager examines if there are any liabilities, e.g. invoices for the actual working period, taking into consideration the defined terms and conditions of payment for each of the invoices.

16 - Zero-account shift manager. The zero-account shift manager examines the remaining possibilities with regard to an optimized cash flow - when it is indicated to charge a liability from an account, where the actual amount on the credit side is not high enough to pay it from one single account.

17 - Cash discount manager. The cash discount manager analyses which liabilities could be met with regard to an offered discount. According to the opportune guidelines the system tries to benefit by paying.

18 - Splitting payment manager. The splitting payment manager takes into consideration that the current amount cannot be paid from one single account without overdraft and selects two or more accounts, in order to split the amount due with regard to an optimized cash flow. The decision - which amount and which accounts are selected - depends mainly on the system preferences, which specify if the recipient accepts a split payment or not.

19 - Rate of interest manager. The rate of interest manager analyses the rates which are due by overdrafts of the accounts and selects the most favourable combination, which not only depends on the rate itself, but also on the actual amount overdraft present on the accounts. In cooperation with the cash discount manager 17 the system decides, if it is opportune, to remit payment at present or to defer it to a subsequent term of payment.

12 - Bank account allocation manager. The bank account allocation manager selects the most opportune account, regarding the actual amount on the credit as well as with regard to the interest rate for overdrafts, in order to start further calculations with the first selected liability.

The above explains operation of the above managers at the data processing level to help describe the context in which the communication steps of the invention are performed. These comments also apply to the descriptions below of the managers within the pattern recognition manager 31.

The pattern assembly manager 11 is expanded out in Fig. 3 and those which use the output of the concepts memory area 10 are illustrated in Fig. 3. There is a transaction manager 21(a) connected to an electronic banking manager 6 which feeds the modem 27.

Referring to Fig. 4, particular detail of the structure file in the concepts memory area is shown for processing of external stimuli or exogenous incitements. The pattern recognition manager 31 comprises:

32 - Bank account report analyser. The bank account report analyser imports the actual statements of the different accounts electronically and starts analysing the contents. Not only the statements documentation is taken into consideration, but also the transferred amount as well as the item's transaction and date specifications. Furthermore, the bank account report analyser starts multidimensional searches in the database files, in order to select the most likely records. When there is a number of more than one possible answers for a certain item, the manager carries out a plausibility control, regarding possible specifications like invoice number, client number, client name, date of invoicing, date of transaction, transferred amount etc.

33 - Debit entry analyser. The debit entry analyser identifies all transactions, of debit or credit to an account according to registered debit notes. The system not only examines if there is a registered legitimation for booking the different debit notes, but also implements different plausibility functions. It checks if the current amount exceeds the usual range, if the term of booking is within the estimated timeframe etc. Furthermore the debit entry analyser induces the bookkeeping process to generate the complete bookkeeping reference. Thereby, the system works according to standard bookkeeping rules.

34 - Money transfer order analyser. The money transfer order analyser works similarly to the debit entry analyser 33, examining the transfer orders which have been arranged with the different banks. Furthermore the money transfer order analyser generates the corresponding complete bookkeeping process by allocating the actual amount to the predefined booking statements.

35 - Cash manager payment report analyser. The cash manager payment report analyser evaluates all processes that debit the accounts as a consequence of transactions which were sent by the system to the different banks according to the cash management calculations one or two days before (compare endogenous incitements).

36 - Failure of assignment manager. The failure of assignment manager controls of all items which debit or credit an account, but could not be allocated by the different managers described above. By using some retrieval procedures, the system analyses if similar or analogous processes have taken place in the past and tries to allocate the corresponding item.

37 - Date of payment manager. The date of payment manager checks all invoices according to the evaluated bank account status and examines if the date of payment corresponds to the date due. If not, the systems induces a reminder process for the amount due.

38 - Terms of payment manager. The terms of payment manager checks all invoices according to the evaluated bank account status and examines if the terms of payment i.e. with regard to discounts etc. are met. If not, the systems induces a reminder process for the remaining amount due.

39 - Multiple payment manager. The multiple payment manager controls if a payment which is credited to an account related to one or more than one invoice. For example, if a client collects several invoices and pays them at once, the system has to evaluate the bank account status and has to check, which invoice records should be addressed within the database. This is especially difficult when there are discounts which have been taken by the client, or when the client pays several invoices at once, and the amount represents only a part of all unpaid invoices.

40 - Cash discount manager. The cash discount manager checks all invoices according to the evaluated bank account status and examines if the client was entitled to take the discount. If not, the system induces a reminder process for the unpaid amount.

41 - Validity reference manager. The validity reference manager tries to identify if a term which has been identified within a bank status report is valid enough to be used as a separator term, for example in order to distinguish between different invoice records.

42 - Part payment manager. The part payment manager recognizes when a liability is partly remitted. The manager has to decide if a reminder process should be initiated, or if the remaining amount due remains within the database for a further clearing procedure.

All of the above modules are part of the pattern recognition manager 31 and feed into the pattern simulation manager 48.

Again, the output from the memory area 10 feeds into speech and text output modules as illustrated in Figs. 2 and 3. However, in addition there is an automatic transaction manager 43(a), a bank account status report manager 43(b), and a data reference update manager 43(c).

As is clear from the above description the expert system of the invention is programmed to preform cash management operations in which decisions are made as to how money should be managed. In each case there will be a number of possible decisions, many of which are generally correct, and the system operates to determine the optimum overall result. The resulting event may be lodgement of an entry to a journal account, transmission of a signal to the bank system for lodgement of money, payment of invoices, or directing printing of cheques for payment of invoices.

The invention resides in understanding inputs and building output communication about the events for users. This is achieved in-line in real time in a dynamic and flexible manner. The output communication is both in text and in speech. An important feature is that as the data is being processed, basic communication elements are processed in combination with linked or associated program code in the form of a concept. In other words, a concept is:-

(a) the basic program code part relating to the fact, matter, result, or any event for which there is a need for communication, and

(b) the corresponding means of communication.

Thus, it will be noted from Fig. 4 that the managers of the pattern recognition manager 31 perform data processing with integrated communication operations using concepts.

In more detail, the basic elements of communication may be single letters, combinations of letters, words, combinations of words, sentences, combinations of sentences, phonetic expressions, numbers, combinations of numbers, figures, or combinations of figures. Generation of concepts in this manner allows communication in both directions. The system coherently generates communication to output information about the events, and it coherently evaluates received data and communicates the results to the user. Any suitable programming language may be used for the program code. The important point is that the generic elements for communication, or alternatively codes which trigger corresponding communication elements, are associated with the corresponding program code to form concepts.

It is preferable that the communication elements are situated within the same memory area. In the specific example of speech, a preferred embodiment is to store the speech resources elsewhere and trigger them with the elements for communication within the concepts memory area, thus allowing speech editing with regard to prosodic control and grammar. In more detail, the speech resources are triggered in a manner whereby they are not loaded in sequence, but instead as blocks of twenty resources at one time. This enables fluent speech and avoids memory overflow. This is achieved by operation of an index table of the speech resources 24 monitoring the process and recognising when a block of resources is nearing its end. It loads the next block and after the new block has started it clears the memory of the previous block. Sentences are formulated with an open end. Thus, they may be of any variable length. A time-dependent handling of the speech in blocks is more practical than byte-related handling as required by the prior art such as EP 0052757B.

The system handles data differently depending on whether or not it originated within the system (endogenous) or externally of the system (exogenous). These differences are illustrated in Figs. 2, 3 and 4. The important point is that the endogenous data does not need to be recognised before it is processed. For the exogenous stimuli the pattern recognition manager 31 performs the necessary recognition functions before pattern simulation and conclusion.

Within the concepts memory 10, the CLAU manager 50 corrects, evaluates, or changes communication built up or recognised by the managers in the memory area 10 (see Fig. 2). For example, the manager 55 will not reject an exogenous incitement which, with the exception of one word, is otherwise coherent with regard to its contents. In other words, it is understood when and if the word in question is a synonym to another which is stored in a concept in the memory area 10. A more comprehensive description of the functions of the managers 51 to 60 which make up the CLAU manager 50 is given above. The important point is that they interact with the expert system managers 11, 20, 31, 48 and 49 to assist in communication understanding and generation.

As stated above, received information is interpreted and output communication is generated in an integrated manner during data processing, using concepts. Communication is understood and built using basic communication elements which should not be confused with pre-defined messages or alerts, because they are used in understanding and building coherent communication in an entirely flexible manner. This communication is purpose-built from the elements of communication linked to the basic program code during data processing with the assistance of the matrix manager 10a. Referring now to the Appendix, samples of the program code are given in which an asterisk indicates examples of the relevant places of linkage.

The first part of the Appendix provides code samples of the pattern assembly manager 11, which recognises inputs. This allows, for example, automatic understanding of who has paid monies, for which reason, and when. On the first page of the Appendix, comments are built up by reference to the splitting payments manager 18. Line B indicates the information used for the coherent communication, and line C activates the speech building process. The Appendix includes comment lines explaining the communication operations in detail.

The matrix manager 10a is an entity which contributes to the building of communication in an online mode. It is a parallel recursive function which is constantly available to the managers with which it operates. Its function is not dependent on the presence of a reference table. This is due to the fact that the elements are by no means limited, as with the prior art. For example, it interacts with the Bank Account Analyser 32 by selecting several elements within the analyser 32 and comparing them to stored data. The plausibility is then checked and if found true all elements analysed will be stored for later output. Accordingly, new elements are stored in a manner which causes the system to learn by widening its knowledge base. The matrix manager 10a may therefore be regarded as a general tool used in the concepts memory area to assist in processing elements or concepts for sorting, comparison, input or output. This is achieved by constantly scrolling through the elements or concepts for communication building.

It will be appreciated that the invention provides a method and system for understanding inputs and generating coherent communication outputs in an entirely flexible manner. Each communication output is entirely new as it is built up from small elements and corresponds coherently to the events which occurred. Because, the communication is built up during processing in an integrated manner, it informs the user not only of occurrence of an event, but also why the event occurred and how the event affects other events. For example, if the event is the fact that no processing result is

possible, then the output communication will specify why there was no result. This is extremely important for users of an expert system.

It will also be appreciated that there is considerable flexibility in communication output. For example, there are no pre-defined endings to sentences - text is added until a coherent communication reflecting the events is achieved. The communication is thus open-ended and this complex information may be communicated at "any length". The only tables which are used are ones storing basic numerals 0 - 9 and the alphabet - communication being dynamically understood and built online.

In summary, the technical communication steps achieve the result of a fluent and coherent output - just like a person explaining the events. For example, a graphical representation of a talking man may be generated on screen while the coherent speech is outputted. The text may be simultaneously displayed.

APPENDIX

Concepts Synthesis within the Pattern Simulation Manager for accounts, endogeneous incitaments

```
c_boolean(◊vAllocated)
◊vAllocated:=False
SEARCH for accounts
    where the balance is greater than 0 before payment of one of the invoices
    where the balance is greater than 0 after the payment of one of the invoices
    where the balance is greater than the predefined minimum after payment
    where the cash flow automatisation status is set to true
    If the number of the according accounts is greater than 0
      ◊vPayCond:="Paying within the positive range"
      ◊vAllocated:=True
    End if
    If ((◊vAllocated=False) & (◊vSpliCancel=False))
      SEARCH for accounts
        where the balance is greater than 0 before the payment of one of the invoices
        where the cash flow automatisation status is set to true
        If (the sum of amounts of the selected accounts>=◊vAmountAkt)
          FK_Prior_Num ("Plus")
          If (◊vAblNight=False)
            If ((◊vFKNumber<=◊vSplNumMax) & (◊vSpliBool=True))
              ◊vPayCond:="Splitting"
              ◊vAllocated:=True
            Else
              ◊vPayCond:=FK_PriorityP1 ($1)
          * A Start Concept Synthesis
            Start Splitting Payment Manager
            Check, if the primary settings for splitting payment are followed
          * B Start the comment building process and add the following terms:
              • the invoice number of the current invoice
              • the offered cash discount
              • Offer the user to switch the current settings according to the needed status
          * C Start the speech building process and add the follow. ref. numbers:
              ArrInsElemStr (»◊ArrSprache;"552")
              `Speech resource 552 standing for the character "a"
```

```
CharAnalyse1 (»0ArrSprache;[BankStatusReport]ID)
  procedure CharAnalyse1 analyses the parameter, which is presented by
  the variable [BankStatusReport]ID. According to the length of the term
  and the digits within the term the procedure CharAnalyse1 appends the
  pointers for the different speech resources (for example Sp.R. 501 for
  digit "1", Sp.R. 552 for character "a" and so on). Every pointer to a
  certain speech resource is added to the speech-array 0aSprache.
ArrInsElemStr (»0ArrSprache;"641")
ArrInsElemStr (»0ArrSprache;"704")
CharAnalyse3 (»0ArrSprache;$Amount)
    procedure CharAnalyse3 analyses a real number, which is presented by
    the variable "$Amount". From digit to digit the procedure assigns a
    speech resource and adds the pointer to this resource to the speech-array
    0aSprache.
ArrInsElemStr (»0ArrSprache;"112")
ArrInsElemStr (»0ArrSprache;"103")
ArrInsElemStr (»0ArrSprache;"101")
  End if
Else
  If ((0vFKNumber<=0vSplNumMax) & (0vSpliBool=True))
    0vPayCond:="Splitting Payment"
    0vAllocated:=True
  End if
End if
End if
End if
If (0vAllocated=False)
Search for all accounts
  where the balance is greater than 0 before the payment of one of the invoices
  where the cash flow automatisation status is set to true
  If the sum of amounts of the selected accounts>=0vAmountAkt)
    FK_Prior_Num ("PlusMinus")
      If (0vAblNight=False)
      Case of
        : (0vFKNumber>1)
          If (0vSpliCancel=False)

        If ((0vFKNumber<=0vSplNumMax) & (0vSpliBool=True))
```

```
            ◊vPayCond:="Splitting Payment"
            ◊vAllocated:=True
         Else
            ◊vPayCond:=FK_PriorityP2 ($1)
      ✳ D Start Concept Synthesis
            Start Splitting Payment Manager
      ✳ E Start the comment building process and add the following terms:
            • the invoice number of the current invoice
            • the offered cash discount
            • the difference between the offered cash discount and the needed
              payment according to the rate of interest, if the invoice shoult be payed
              now
      ✳ F Start the speech building process and add the follow. ref. numbers:
            ArrInsElemStr (»◊ArrSprache;"604")
            `Speech resource 552 standing for the partial word "Split"
            CharAnalyse1 (»◊ArrSprache;[BankStatusReport]ID)
             procedure CharAnalyse1 analyses the parameter, which is presented by
             the variable [BankStatusReport]ID. According to the length of the term
             and the digits within the term the procedure CharAnalyse1 appends the
             pointers for the different speech resources (for example Sp.R. 501 for
             digit "1", Sp.R. 552 for character "a" and so on). Every pointer to a
             certain speech resource is added to the speech-array ◊aSprache.
            ArrInsElemStr (»◊ArrSprache;"707")
            ArrInsElemStr (»◊ArrSprache;"115")
            CharAnalyse3 (»◊ArrSprache;$Difference)
              procedure CharAnalyse3 analyses a real number, which is presented by
              the variable "$Difference". From digit to digit the procedure assigns a
              speech resource and adds the pointer to this resource to the speech-array
              ◊aSprache.
            ArrInsElemStr (»◊ArrSprache;"127")
            ArrInsElemStr (»◊ArrSprache;"895")
            ArrInsElemStr (»◊ArrSprache;"222")
         End if
      End if
   : (◊vFKNumber=1)
      ◊vPayCond:="Splitting Payment"
      ◊vAllocated:=True
End case
```

For cases when exogenous incitaments prevail the algorithm of the Pattern Recognition Manager for accounts is demonstrated.

Concepts Synthesis within the Pattern Recognition Manager for accounts, exogeneous incitaments

```
C_TEXT(◊vComment)
C_LONGINT($i)
◊vComment:=""
USE SET("BankStatusReportSet")
FIRST RECORD([BankStatusReport])
RecLoadWait (»[BankStatusReport])
For ($i;1;Records in selection([BankStatusReport]))
 [BankStatusReport]DocuText:=SpaceDelete ([BankStatusReport]DocuText)
 Case of
  : ([BankStatusReport]Amount>0)
   If (AuszugFilterRA ("NichtBeNumt")=True)
    [BankStatusReport]Detected:=True
    SAVE RECORD([BankStatusReport])
   Else
    If (AuszugFilterZE =True)
     [BankStatusReport]Detected:=True
     SAVE RECORD([BankStatusReport])
    Else
     If (AuszugFilterRA ("BeNumt")=True)
      [BankStatusReport]Detected:=True
      SAVE RECORD([BankStatusReport])
     Else
      Start Concept Synthesis
      Start the comment building process and add the following terms:
        If (Length(◊vComment)#0)
          ◊vComment:=◊vComment+◊CR+◊CR
```

```
        End if
        ◊vComment:=◊vComment+"Filtervorgang: Bankauszug Nummer " +[BankStatus
          Report]ID+":"+◊CR
        ◊vComment:=◊vComment+"Eine Zuordnung ist bislang noch nicht möglich!"
      *I Start the speech building process and add the follow. ref. numbers:
        ArrInsElemStr (»◊ArrSprache;"552")
        CharAnalyse1 (»◊ArrSprache;[BankStatusReport]ID)
        ArrInsElemStr (»◊ArrSprache;"553")
       End if
      End if
     End if
    : ([BankStatusReport]Amount<0)
     If (StatusFiltDebit =True)
      [BankStatusReport]Detected:=True
      SAVE RECORD([BankStatusReport])
     Else
      If (StatusFilterZA =True)
        [BankStatusReport]Detected:=True
        SAVE RECORD([BankStatusReport])
      Else
      *J Start Concept Synthesis
      *K Start the comment building process and add the following terms:
        If (Length(◊vComment)#0)
          ◊vComment:=◊vComment+◊CR+◊CR
        End if
        ◊vComment:=◊vComment+"Filtervorgang: Bankauszug Nummer "+[BankStatus
        Report]ID+":"+◊CR
        ◊vComment:=◊vComment+"Eine Zuordnung ist bislang nicht möglich!"
      *L Start the speech building process and add the follow. ref. numbers:
        ArrInsElemStr (»◊ArrSprache;"552")
        CharAnalyse1 (»◊ArrSprache;[BankStatusReport]ID)
        ArrInsElemStr (»◊ArrSprache;"553")
       End if
     End if
    End case
    NEXT RECORD([BankStatusReport])
    RecLoadWait (»[BankStatusReport])
   End for
```

The speech resources were stored in resources and subsequently triggered by arrays. This is demonstrated by the following algorithm:

```
Communication Matrix Manager
case of
: (var=digit)
 C_POINTER($1)
 C_STRING(80;$2)
 Case of
  : (Ascii($2≤(Length($2)-7)≥)=49)
   ArrInsElemStr ($1;"1000")
  : (Ascii($2≤(Length($2)-7)≥)>49)
   ArrInsElemStr ($1;$2≤(Length($2)-7)≥)
   ArrInsElemStr ($1;"2000")
 End case
 Case of
  : (Ascii($2≤(Length($2)-5)≥)=49)
   ArrInsElemStr ($1;"328")
   ArrInsElemStr ($1;"100")
  : (Ascii($2≤(Length($2)-5)≥)>49)
   ArrInsElemStr ($1;$2≤(Length($2)-5)≥)
   ArrInsElemStr ($1;"100")
 End case
 Case of
  : ((Ascii($2≤(Length($2)-4)≥)#48) & (Ascii($2≤(Length($2)-3)≥)=48))
   ArrInsElemStr ($1;Substring($2;(Length($2)-4);2))
  : ((Ascii($2≤(Length($2)-4)≥)#48) & (Ascii($2≤(Length($2)-3)≥)#48))
   ArrInsElemStr ($1;Substring($2;(Length($2)-4);2))
  : ((Ascii($2≤(Length($2)-4)≥)=48) & (Ascii($2≤(Length($2)-3)≥)#48))
   ArrInsElemStr ($1;$2≤(Length($2)-3)≥)
 End case
 Case of
```

```
    : ((Ascii($2≤(Length($2)-1)≥)#48) & (Ascii($2≤Length($2)≥)=48))
      ArrInsElemStr ($1;"101")
      ArrInsElemStr ($1;Substring($2;(Length($2)-1);2))
    : ((Ascii($2≤(Length($2)-1)≥)=48) & (Ascii($2≤Length($2)≥)#48))
      ArrInsElemStr ($1;"101")
      ArrInsElemStr ($1;"0")
      ArrInsElemStr ($1;$2≤Length($2)≥)
    : ((Ascii($2≤(Length($2)-1)≥)#48) & (Ascii($2≤Length($2)≥)#48))
      ArrInsElemStr ($1;"101")
      ArrInsElemStr ($1;Substring($2;(Length($2)-1);2))
  End case
  :(var = single character)
  C_LONGINT($i)
  C_POINTER($1)
  C_STRING(80;$2)
  For ($i;1;Length($2))
   Case of
    : ((Ascii($2≤$i≥)>=48) & (Ascii($2≤$i≥)<=57))
      ArrInsElemStr ($1;$2≤$i≥)
    : ((Ascii($2≤$i≥)>=65) & (Ascii($2≤$i≥)<=90))
      ArrInsElemStr ($1;String(Ascii($2≤$i≥)+536))
    : ((Ascii($2≤$i≥)>=97) & (Ascii($2≤$i≥)<=122))
      ArrInsElemStr ($1;String(Ascii($2≤$i≥)+504))
   End case
  End for
  :(var=date)
  C_POINTER($1)
  C_STRING(80;$2;$Zahl)
  C_LONGINT($i)
  C_BOOLEAN($Punkt)
  $Punkt:=False
  $i:=1
  $Zahl:=""
  Repeat
```

```
If ($2≤$i≥=".")
 $Punkt:=True
Else
 $Zahl:=$Zahl+$2≤$i≥
 $i:=$i+1
End if
Until ($Punkt=True)
ArrInsElemStr ($1;String(Num($Zahl)+500))
$Punkt:=False
$i:=$i+1
$Zahl:=""
Repeat
 If ($2≤$i≥=".")
  $Punkt:=True
 Else
  $Zahl:=$Zahl+$2≤$i≥
  $i:=$i+1
 End if
Until ($Punkt=True)
ArrInsElemStr ($1;String(Num($Zahl)+500))
ArrInsElemStr ($1;String(Num(Substring($2;(Length($2)-3);4))-1462))
Q :(var=word)
C_POINTER($1)
C_STRING(80;$2)
INSERT ELEMENT($1»;Size of array($1»)+1;1)
$1»{Size of array($1»)}:=$2
end case
```

Claims

1. A communication method for interpreting received communication and building output coherent communication indicating system events, the method being carried out by a system comprising a processor (47, 8), a memory (3, 5, 10), an input device (9, 29), and an output device (28, 25, 26), characterised in that :

   the processor (47, 8) automatically interprets inputs or generates coherent communication in-line as data is being processed using a communication means linked with data processing program code.

2. A method as claimed in claim 1, characterised in that the processor (47) processes concepts to interpret inputs or generate coherent communication, each concept comprising a communication element linked to a part of program code.

3. A method as claimed in claim 2, characterised in that the concepts are processed in one memory area (10), either actual or virtual, and either time-dependent or time-independent.

4. A method as claimed in claims 2 or 3, characterised in that the processor processes concepts having communication elements linked with program code representing processing results, and concepts having communication elements linked with program code used for interim processing before a result is determined.

5. A method as claimed any preceding claim, characterised in that the processor interactively utilises a communication matrix manager (109) providing a parallel recursive function for data comparison to build a communication knowledge base.

6. A method as claimed in any preceding claim, characterised in that the processor (47, 8)uses a conception, learning, associative and utilisation manager (50) to perform communication operations said manager including:-

- a definition of terms manager (51) which performs a full text analysis of inputs to define as many terms as possible, and

- a validation manager (52) which performs validation operations on the defined terms.

7. A method as claimed in any preceding claim, characterised in that the communication means comprises communication elements which trigger resources (24) including text or speech strings.

8. A method as claimed in claim 7, characterised in that the system comprises speech resources (24), a speech resources assembly manager (21(c)), and a speech resources reference buffer (23(b)), the assembly manager and the reference buffer performing the steps of:-

- loading blocks of resources to a memory;

- monitoring speech output and determining when resources of a block are nearly utilised;

- loading a fresh block; and

- clearing the previous block when a fresh block is activated.

9. A method as claimed in claim 7, characterised in that the resources are used to build coherent output communication which is further automatically edited (23(e), 23(d)) before output.

10. A data processing system comprising a processor (47, 8), a memory (3, 10, 5), an input device (9, 29), and an output device (28, 25, 26), characterised in that the processor (47, 8) comprises means for automatically interpreting inputs or generating coherent communication as data is being processed using a communication means linked with data processing system code.

FIG 1

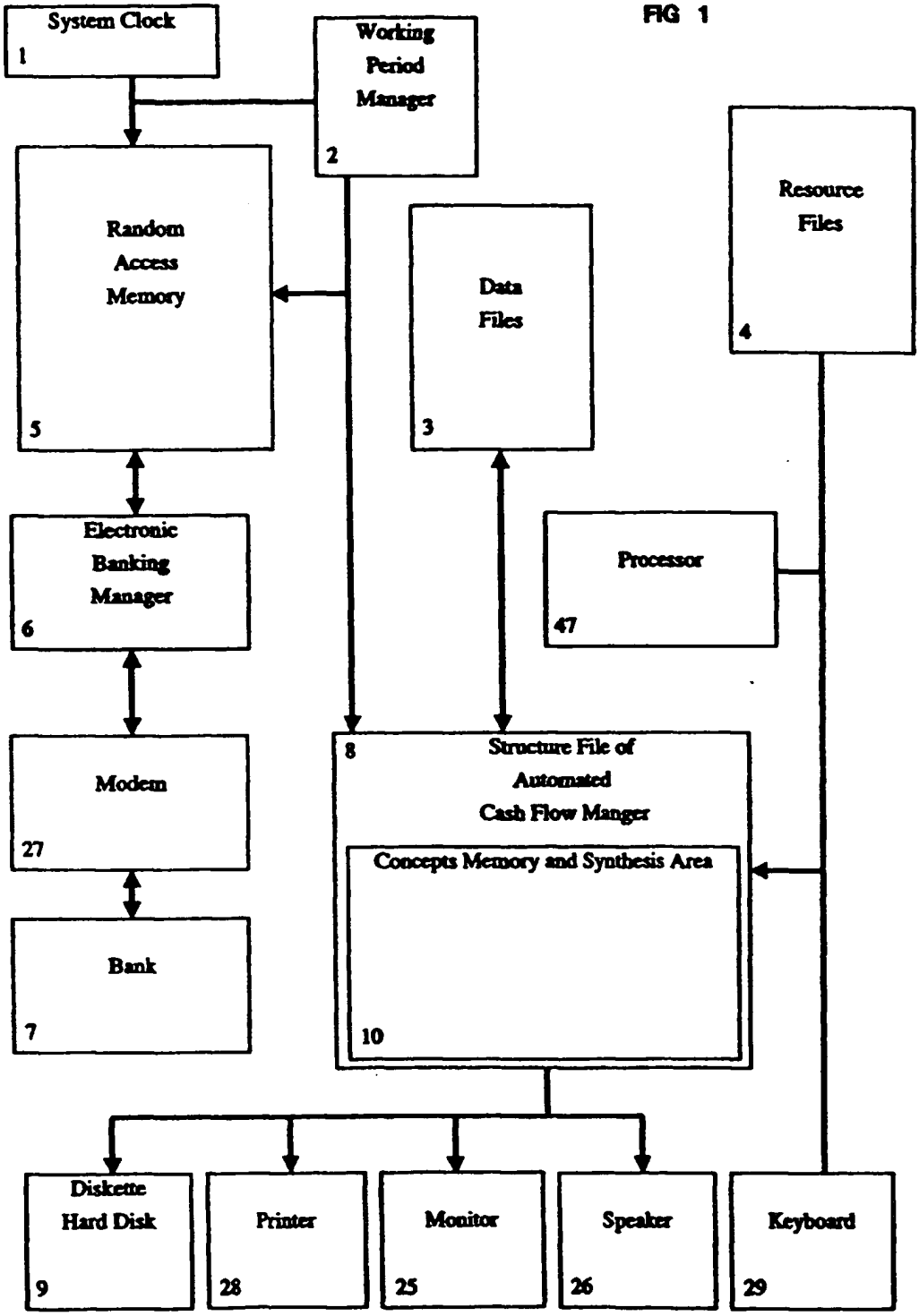

System Clock
1
Working Period Manager
2
Random Access Memory
5
Data Files
3
Resource Files
4
Electronic Banking Manager
6
Processor
47
Modem
27
8
Structure File of Automated Cash Flow Manger
Concepts Memory and Synthesis Area
10
Bank
7
Diskette Hard Disk
9
Printer
28
Monitor
25
Speaker
26
Keyboard
29

FIG 2

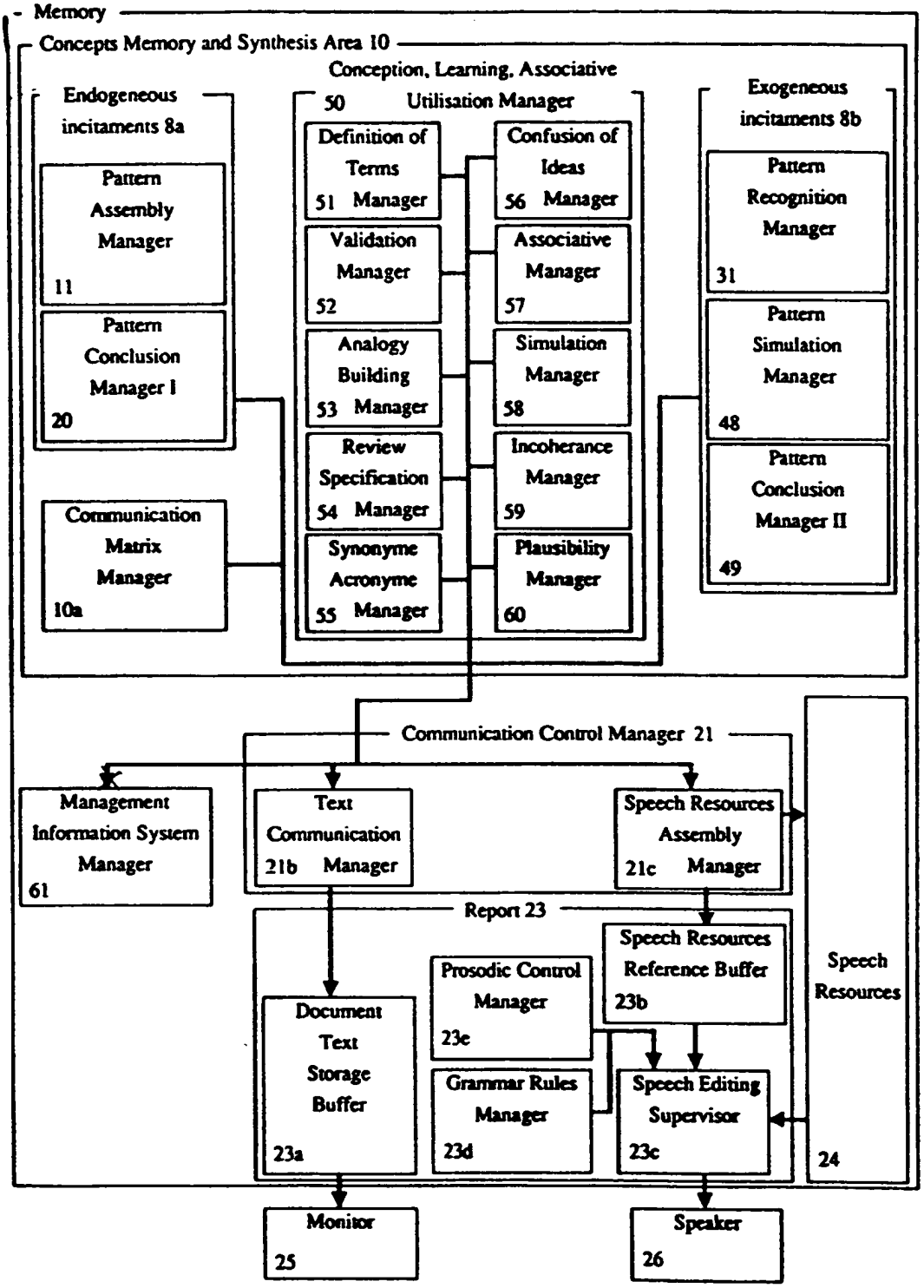

Memory
Concepts Memory and Synthesis Area 10
Conception, Learning, Associative
Endogeneous incitaments 8a
Pattern Assembly Manager 11
Pattern Conclusion Manager I 20
Communication Matrix Manager 10a
50 Utilisation Manager
Definition of Terms 51 Manager
Validation Manager 52
Analogy Building 53 Manager
Review Specification 54 Manager
Synonyme Acronyme 55 Manager
Confusion of Ideas 56 Manager
Associative Manager 57
Simulation Manager 58
Incoherance Manager 59
Plausibility Manager 60
Exogeneous incitaments 8b
Pattern Recognition Manager 31
Pattern Simulation Manager 48
Pattern Conclusion Manager II 49
Communication Control Manager 21
Management Information System Manager 61
Text Communication 21b Manager
Speech Resources Assembly 21c Manager
Report 23
Document Text Storage Buffer 23a
Prosodic Control Manager 23e
Grammar Rules Manager 23d
Speech Resources Reference Buffer 23b
Speech Editing Supervisor 23c
Speech Resources 24
Monitor 25
Speaker 26

FIG 3

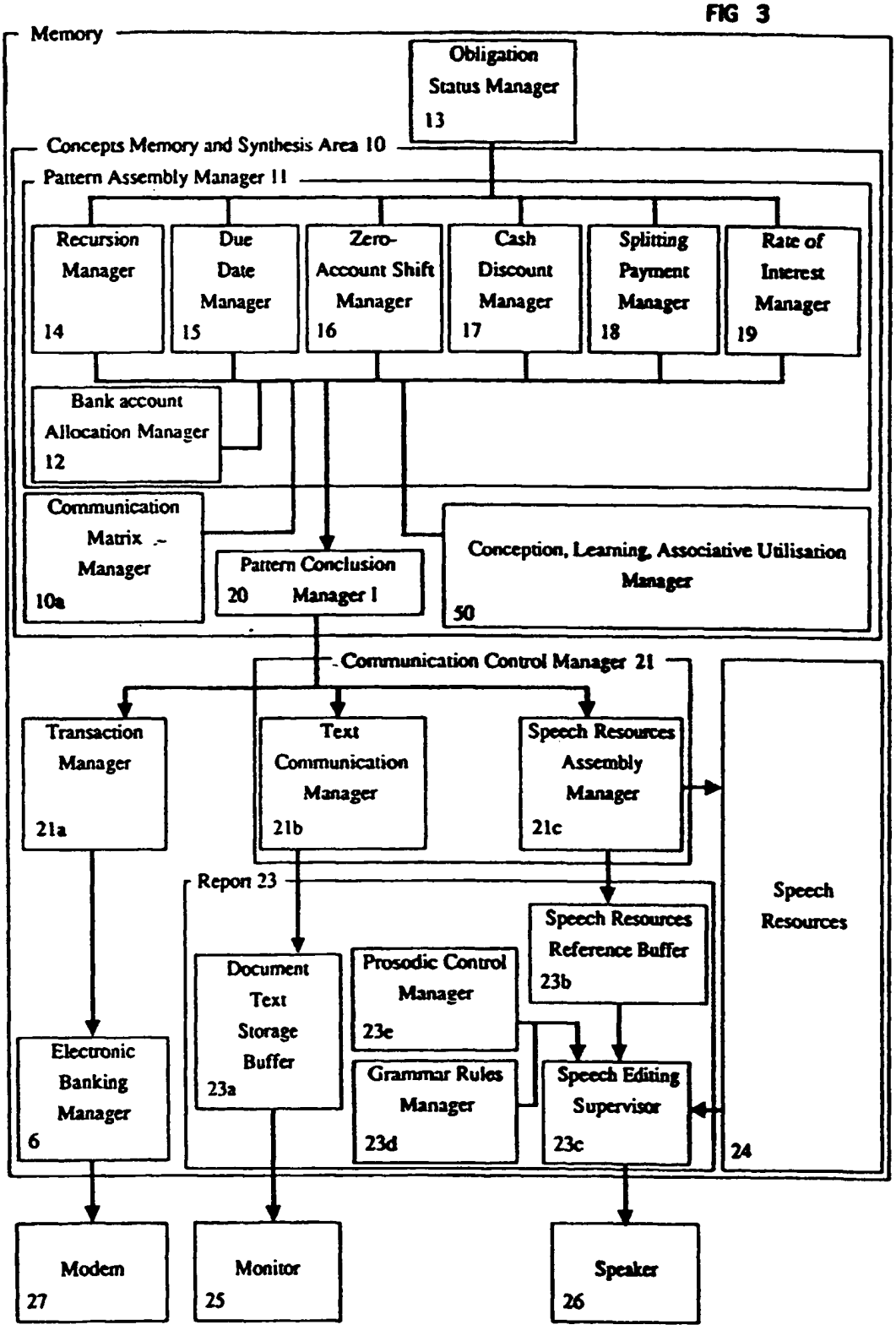
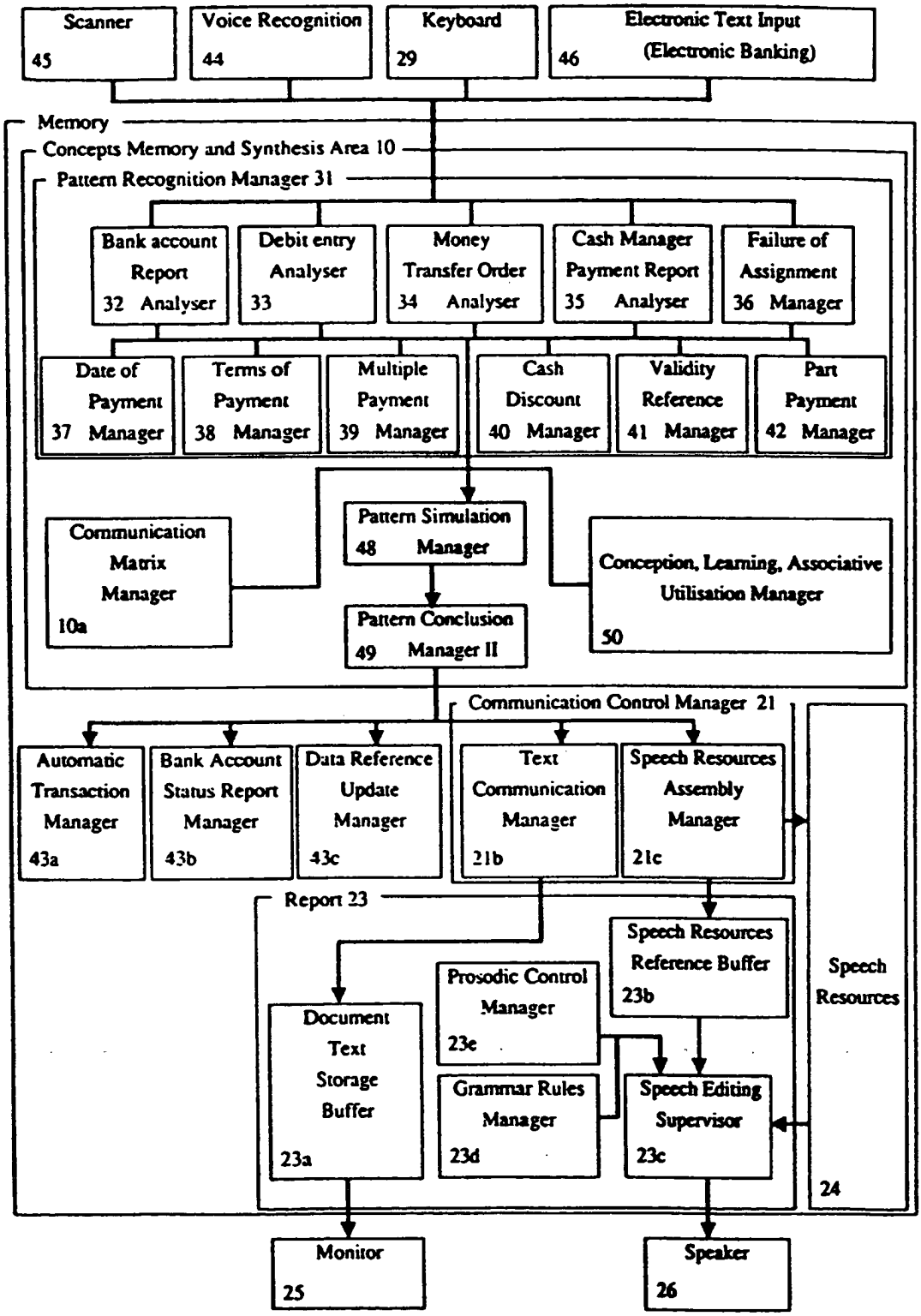

Memory
Obligation Status Manager 13
Concepts Memory and Synthesis Area 10
Pattern Assembly Manager 11
Recursion Manager 14
Due Date Manager 15
Zero-Account Shift Manager 16
Cash Discount Manager 17
Splitting Payment Manager 18
Rate of Interest Manager 19
Bank account Allocation Manager 12
Communication Matrix Manager 10a
Pattern Conclusion 20 Manager I
Conception, Learning, Associative Utilisation Manager 50
Communication Control Manager 21
Transaction Manager 21a
Text Communication Manager 21b
Speech Resources Assembly Manager 21c
Report 23
Speech Resources
Speech Resources Reference Buffer 23b
Document Text Storage Buffer 23a
Prosodic Control Manager 23e
Grammar Rules Manager 23d
Speech Editing Supervisor 23c
24
Electronic Banking Manager 6
Modem 27
Monitor 25
Speaker 26

FIG 4


Scanner
45
Voice Recognition
44
Keyboard
29
Electronic Text Input
46
(Electronic Banking)
Memory
Concepts Memory and Synthesis Area 10
Pattern Recognition Manager 31
Bank account Report 32 Analyser
Debit entry Analyser 33
Money Transfer Order 34 Analyser
Cash Manager Payment Report 35 Analyser
Failure of Assignment 36 Manager
Date of Payment 37 Manager
Terms of Payment 38 Manager
Multiple Payment 39 Manager
Cash Discount 40 Manager
Validity Reference 41 Manager
Part Payment 42 Manager
Communication Matrix Manager 10a
Pattern Simulation 48 Manager
Pattern Conclusion 49 Manager II
Conception, Learning, Associative Utilisation Manager 50
Communication Control Manager 21
Automatic Transaction Manager 43a
Bank Account Status Report Manager 43b
Data Reference Update Manager 43c
Text Communication Manager 21b
Speech Resources Assembly Manager 21c
Report 23
Document Text Storage Buffer 23a
Prosodic Control Manager 23e
Grammar Rules Manager 23d
Speech Resources Reference Buffer 23b
Speech Editing Supervisor 23c
Speech Resources 24
Monitor 25
Speaker 26

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 96 65 0050

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
Y	EP 0 720 090 A (CANON KABUSHIKI KAISHA) * the whole document * ---	1-10	G06F17/28
Y	EP 0 720 106 A (CANON KABUSHIKI KAISHA) * the whole document * ---	1-10	
D,A	EP 0 052 757 A (INTERNATIONAL BUSINESS MACHINES CORPORATION) * the whole document * ---	1,10	
D,A	PATENT ABSTRACTS OF JAPAN vol. 96, no. 4, 30 April 1996 & JP 07 319538 A (NISSIN ELECTRIC CO LTD), 8 December 1995, * abstract * -----	1,10	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)

G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
BERLIN	6 March 1997	Abram, R

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)